Міністерство освіти і науки України

Національний технічний університет України «Київський політехнічний

інститут імені Ігоря Сікорського"

Факультет інформатики та обчислювальної техніки

Кафедра інформатики та програмної інженерії

Звіт

з лабораторної роботи № 5 з дисципліни

«Основи програмування 1. Базові конструкції»

Варіант 25

Виконав студент ІП-14 Радзівіло Валерія Артемівна

(шифр, прізвище, ім'я, по батькові)

Перевірив

( прізвище, ім'я, по батькові)

Київ 2021

**Лабораторна робота 5**

**Організація циклічних процесів. Складні цикли**

**Мета:** вивчити особливості організації складних циклів.

**Задача**:



**Розв’язання**

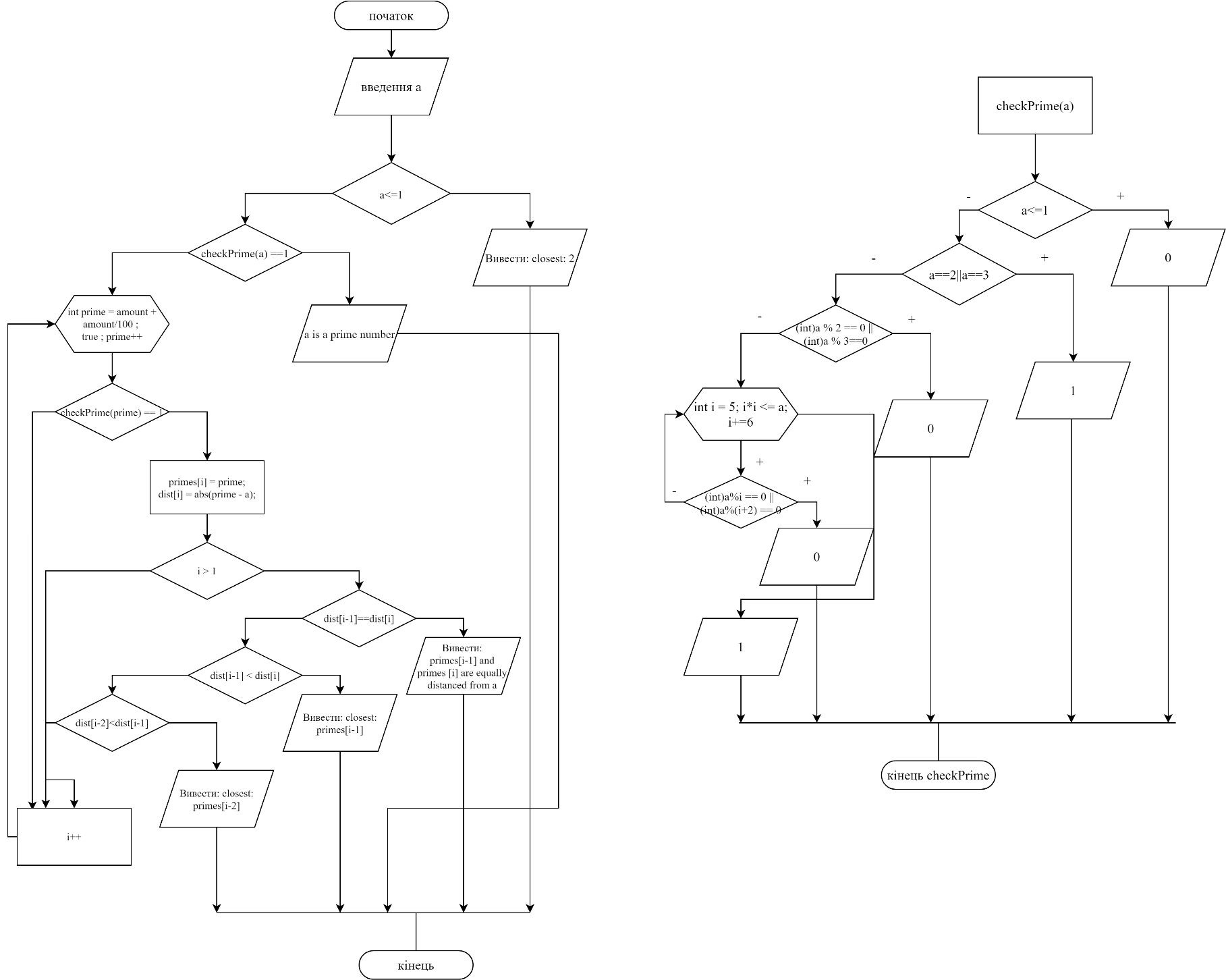
Програмні специфікації зазначимо у графічній формі у вигляді блок-схеми.

**Математична модель**

|  |  |  |  |
| --- | --- | --- | --- |
| **Змінна** | **Тип** | **Ім’я** | **Призначення** |
| Функція checkPrime | boolean | checkPrime | Функція |
| Ціле число a | Цілий | a | Частина функції |
| Дійсне число a | Дійсний | а | Початкове данне |
| Ціле число amount | Цілий | amount | Проміжкове дане |
| Цілочисельний масив primes | Масив | primes | Проміжкове дане |
| Цілочисельний масив dist | Масив | primes | Проміжкове дане |
| Ціле число prime | Цілий | prime | Проміжкове дане |
| Ціле число i | Цілий | i | Проміжкове дане |
| Ціле число з масиву primes | Цілий | primes[i] | Результат |

**Крок 1**

**Крок 2**

****

**Вирішення задачі С++:**

#include <iostream>

#include <cmath>

using namespace std;

bool checkPrime(int a)

{

    if(a<=1)

    {

        return 0;

    }

    if(a==2||a==3)

    {

        return 1;

    }

    if( (int)a % 2 == 0 || (int)a % 3==0)

    {

        return 0;

    }

    else{

        for(int i = 5; i\*i <= a; i+=6)

        {

            if((int)a%i == 0 || (int)a%(i+2) == 0)

            {

                return 0;

            }

        }

        return 1;

    }

}

int main()

{

    float a;

    cout << "Enter a number: ";

    cin >> a;

    int amount = (int)a/2;

    int primes[amount+2];

    float dist[amount+2];

    if(a<=1)

    {

        cout << "closest: 2" << endl;

        exit;

    }

    if(checkPrime(a) == 1)

    {

        cout << a <<" is a prime number" << endl;

        exit;

    }

    else{

    int i = 0;

    for(int prime = amount + amount/100 ; true ; prime++)

    {

        if(checkPrime(prime) == 1)

        {

            primes[i] = prime;

            dist[i] = abs(prime - a);

        if(i>1)

        {

            if(dist[i-1]==dist[i])

        {

            cout << primes[i-1] << " and " << primes [i] << " are equally distanced from " << a << endl;

            return 0;

        }

        else{

        if(dist[i-1] < dist[i])

        {

           cout <<"closest: " << primes[i-1] << endl;

           return 0;

        }

         if(dist[i-2]<dist[i-1])

        {

            cout <<"closest:" << primes[i-2] << endl;

            return 0;

        }

    }

        }

         i++;

        }

    }

    }

    }

**Результат:**

****

**Python:**

def checkPrime(a):

    if a <=1:

        return 0

    if a == 2 or a==3:

        return 1

    if int(a) % 2 ==0 or int(a) % 3 ==0:

        return 0

    else:

        i = 5

        while i\*i<=a:

            if int(a)%i == 0 or int(a)%(i+2) == 0:

                return 0

            i+=6

        return 1

a = float(input("Enter a number: "))

amount = int(a)/2

primes=[]

dist=[]

if a<=1:

    print("closest: 2")

    exit()

if checkPrime(a)==1:

    print(a, " is a prime number")

    exit()

else:

    i = 0

    prime = int(amount + amount/100)

    while True:

        if checkPrime(prime)==1:

            primes.insert(i,prime)

            dist.insert(i, abs(prime - a))

            if i >1:

                if dist[i-1] == dist[i]:

                    print(primes[i-1], " and ", primes[i], " are equally distanced from ", a)

                    exit()

                else:

                    if dist[i-1] < dist[i]:

                        print("closest: ", primes[i-1])

                        exit()

                    if dist[i-2] < dist [i-1]:

                        print("closest: ", primes[i-2])

                        exit()

            i=i+1

        prime = prime+1

**Результат:**

****

**Висновок**: У цій лабораторній роботі було вивчено особливості організації складних циклів. Була постановлена задача, в якій визначалося число, введене користувачем. У алгоритмі були уточнені обмеження з використанням умов при яких задача буде виконана правильно. За допомоги цієї лабораторної роботи можливо обчислення найближчих простих чисел.